VISIT…

LANZAROTE Caliente.COM

Материалы III Всеросс. конф. "Математика, информатика, управление – 2004". Иркутск, 2004.

ПЕРСПЕКТИВЫ РАСПАРАЛЛЕЛИВАНИЯ ПРОГРАММ НЕЙРОСЕТЕВОГО АНАЛИЗА И ОБРАБОТКИ ДАННЫХ

Царегородцев В.Г.

www.NeuroPro.ru tsar@neuropro.ru

При создании параллельных программ-нейроимитаторов предлагается в первую очередь распараллеливать не обучение единичной нейросети, а сервисные процедуры: методы оптимизации предобработки данных, схемы отбора примеров в обучающую выборку, отбор информативных независимых переменных, гибридные методы (строящие коллектив экспертов и правила разделения фрагментов задачи между ними) и т.д.. Обучение единичной нейросети может быть распараллелено только в рамках парадигмы растущих сетей или при использовании отдельных методов обучения.

Введение

Оцениваются современные перспективы создания параллельных программ-имитаторов искусственных нейронных сетей. В качестве основной рассматриваемой в работе нейроархитектуры взяты нейронные сети, обучаемые алгоритмами обратного распространения ошибки и решающие задачи обучения с учителем [1]. Традиционное мнение на основе недостаточной производительности стандартных нейропрограмм (Statistica Neural Networks, Trajan, NeuroSolutions и др.) в первую очередь предлагает для распараллеливания задачу обучения нейросети.

Опыт автора и оценка современных научных достижений в нейроинформатике говорят о желательности реализации и распараллеливания специальных сервисных процедур вместо распараллеливания алгоритмов обучения единичной нейросети.

Настоящее обращение к рассмотрению вопросов параллельной реализации нейропрограмм основано в основном на следующих факторах:

1. Пользователю проще использовать программы-нейроимитаторы для обычных компьютеров, чем специализированное аппаратное обеспечение нейровычислений, которое часто требует программирования на более низком уровне, чем программирование для современных универсальных компьютеров, и имеет пиковую производительность как правило на достаточно узком круге задач или при определенных свойствах обрабатываемых данных.
2. Развитие средств кластеризации компьютеров и упрощение практики программирования под кластерные платформы позволяет быстро и дешево организовывать высокопроизводительные и одновременно универсальные вычислительные мощности. А технология hyper threading современных вариантов процессоров Intel Pentium IV дает и пользователю единичного компьютера возможность двухпоточного исполнения программ.
3. Пусть в базовое "нейроядро" нейропрограммы включены только процедуры функционирования сети, оценки качества решения, обратного распространения ошибки и коррекции адаптивных весовых коэффициентов сети. Пусть логика алгоритмов обучения (схема организации перебора примеров обучающей выборки, условие останова обучения, стратегия адаптации нейросети – on-line или batch-стратегия, способ коррекции направления градиента) рассматривается как внешний к базовому ядру блок при проектировании внутренней структуры нейропрограммы. Скорость работы (в последовательном, не параллельном режиме) нейроядра и, соответственно, нейропрограммы доводится практически до максимума при

представлении наборов сигналов и переменных нейросети как векторов и матриц и использованием расширений системы команд процессоров Intel (наборы команд SSE, SSE2 – только векторно-матричная арифметика позволит их с пользой задействовать). Дальнейшее увеличение общей скорости работы может быть достигнуто только реализацией иных алгоритмов (для алгоритмов обучения возможности этого будут рассмотрены в следующем разделе).

Простота достижения предела скорости нейроядра как базового способа повышения эффективности и малые возможности распараллеливания нейроядра делают необходимым распараллеливание преимущественно сервисных алгоритмов. У каждого из профессиональных пользователей нейротехнологий набор сервисных алгоритмов может быть своим – просто потому, что разные люди решают разные задачи, разными методами и при разных требованиях к виду, точности и свойствам решения. Поэтому настоящая работа перечисляет только несколько групп сервисных задач и алгоритмов и показывает возможность их параллельной реализации.

Задача обучения единичной нейросети

Коррекция адаптивных внутренних параметров нейросети может идти как после просмотра каждого очередного примера обучающей выборки, так и путем накопления градиентов всех примеров и использования суммарного антиградиента. Вторая схема может быть легко распараллелена, но теоретически и практически проигрывает по достигаемой скорости обучения сети online-обучению [2,3]. Суммарный градиент, особенно при его дополнении эффективными методами оптимизации типа сопряженных градиентов, часто приводит к переобучению сети [4,5]. Поэтому для предотвращения побочных эффектов лучше использовать online-обучение, которое не распараллеливается, а распараллеливать для него, например, схему [5] начального определения оптимального шага коррекции, дающего высокую скорость обучения наравне с достижением высоких обобщающих способностей сети.

Online-обучение мало чувствительно к величине шага: в [6] высокая скорость сходимости наблюдалась для величин шагов из интервала длиной в два порядка, а замедление сходимости или расхождение наблюдались за пределами этого интервала. А для batch-обучения данные [7] говорят, что оптимум может быть очень узким и лежать внутри одного порядка величины, т.е. при экспериментальном определении длины шага требуется очень детальное сканирование возможного, и поэтому заведомо расширенного интервала значений, что увеличивает требуемый объем рассчетов.

Поэтому реально выигрыш от распараллеливания процесса обучения единичной нейросети можно получить только в случае большого размера обучающих выборок (десятки и сотни тысяч примеров – тогда параллельный расчет суммарного градиента может привести к сокращению общего времени обучения) или при адаптации шага на каждой итерации (эпохе) обучения, когда можно параллельно выполнить пробы нескольких значений величин шагов.

Предобработка данных, выбор примеров в обучающую выборку

Скорость обучения очень сильно зависит от схемы предобработки данных перед их подачей нейросети [8]. Вычисление константы Липшица обучающей выборки как индикатора оптимальности предобработки [8] может быть распараллелено.

Для задач классификации с учителем и неперекрывающихся плотностях распределений примеров разных классов можно отобрать примеры, лежащие вдоль разделяющей классы поверхности, распараллеливаемым методом [9], а далее на меньшем числе примеров и оптимизировать предобработку, и обучать нейросеть. Но задачи с неперекрывающимися плотностями распределения классов, т.е. с

теоретической возможностью получения 100-процентного качества последующего распознавания тестовых выборок, встречаются на практике редко. Да и сложно построить точные индикаторы такой ситуации.

Но есть и другие возможности и стратегии обучения на подмножестве обучающей выборки. Например, селекция [10] после каждой итерации обучения очередного малого числа обучающих примеров. А в [11] предлагается не фиксировать размер обучающей выборки (или соотношение размеров обучающей и тестовой выборок), а начинать с некоторой малой обучающей выборки и после обучения на ней нескольких нейросетей переносить в нее из тестовой выборки примеры с систематической ошибкой (которые неправильно решаются всеми или большинством сетей), после чего обучать новые сети на увеличенной выборке и снова исследовать систематическую ошибку. Используемые в этих случаях этапы тестирования всей массы примеров или обучения нескольких нейросетей могут быть распараллелены.

Широко востребованы также и методы поиска и исправления «нетипичных» примеров (выбросов) в данных, даже при робастных схемах обучения нейросети [12]. Любая робастная схема имеет гипотезы о природе нетипичности, обоснованность которых для конкретной выборки не всегда удается подтвердить. Да и часто не удается применить робастные схемы. Так, способ [13] кластеризации выборки и последующего проецирования примеров, выходящих за границу типичного разброса внутри кластера, ближе к центру кластера не всегда применим: для небольших выборок или большого числа "шумовых" признаков надежная кластеризация затруднена, а евклидово расстояние как мера близости объектов ненадежно при большом числе булевых или номинальных признаков. Авторский способ [14] повышения надежности решения обратной задачи путем применения решающей прямую задачу сети-классификатора также включает алгоритм [13] квантования и исправления данных, поэтому в качестве замены кластеризации методом динамических ядер, используемой в [13], интересны и другие, более мощные и гибкие схемы кластер-анализа.

На этапах предобработки данных, отбора подмножеств примеров, поиска нетипичных наблюдений, при разведочном анализе данных повышение скорости работы алгоритмов (через их распараллеливание) позволит проводить более быстрый, глубокий и всесторонний анализ свойств данных.

Схемы экстраполяции и оценивания предельных значений эффектов

Рассмотрение графиков точностей решения обучающих и тестовых выборок вдоль оси размера обучающей выборки [15] или оси размера нейросети [16] позволяет определить минимально необходимый размер обучающей выборки, с которого заканчивает проявляться чувствительность качества решения к размеру выборки, или максимальный размер сети, с ростом которого начинает проявляться эффект "переобучения" ("запоминания" свойств обучающей выборки с одновременным снижением качества обобщения). Оценка методом [16] нужна и при недоступности регуляризующих процедур при обучении, поскольку последние могут ликвидировать эффект переобучения в нейросетях избыточного размера (но, в свою очередь, требуют подбора веса вклада дополнительного слагаемого в оптимизируемой целевой функции). В рамках идей [15,16] можно получить и оценку уровня шума или уровня противоречивости выборки, ниже которого среднюю ошибку решения новых примеров снизить не удастся. В [17] показано, что из нескольких рассмотренных интер- и экстраполирующих зависимостей одна является стабильно наиболее точной, что подтверждает приемлемость идей [15,16] для возможности прогноза.

Задача обучения нескольких нейросетей для последующего принятия решения о выборе тех или иных настроек является естественным кандидатом на параллельную

реализацию. Она практически в неизменном виде возникла в разделе, обсуждающем алгоритмы обучения, и будет возникать в следующих разделах, поэтому больше акцентировать внимания на ней не будем, а опишем только содержательно отличные внешние смысловые рамки и задачи, где она решается как подзадача.

Растущие нейронные сети, коллективы нейросетей

Распараллеливание обучения единичной сети возможно в схемах подбора размера сети одновременно с ее обучением – путем старта с минимального размера и добавления нейронов в сеть по-одному. Начиная с работы [18] эти схемы могут параллельно пробовать варианты помещения нейронов в разные места сети (в один из имеющихся слоев или в новый слой), либо варианты нейронов с разными нелинейными функциями (номенклатура нелиненых функций дана в [19]) и т.д. и затем выбирать и реально встраивать в сеть только один нейрон с наилучшим критерием качества.

Может быть параллельно обучен набор сетей для последующего объединения в комитет. Так, для задачи классификации с большим числом классов в [20] ведётся обучение отдельной сети для каждой возможной задачи парной классификации. Кроме того, что каждая из сетей будет обучаться на выборке гораздо меньшего размера, чем исходная выборка с примерами для всех классов, возможна оптимизация предобработки данных для каждой сети, а получаемая информация о точности попарного разделения классов может помочь в осмыслении свойств задачи. Для каждой сети можно оптимально подобрать размер (в [20] сеть растет с минимального размера до необходимого), исключить неинформативные признаки (которые могут негативно влиять на качество решения и его чувствительность к шумам). Реально же для задачи построения комитетов нейросетей предложено большое число подходов, доказавших свою эффективность и имеющих разные полезные свойства.

Гибридные алгоритмы

Гибридные алгоритмы используют два и более метода, не обязательно нейросетевых, для построения иерархического многошагового решения: первый метод строит разделение областей компетенции, а второй – решающую модель для области. Так, итерационный алгоритм [21] сжатия данных грубо квантует данные методом динамических ядер и затем более точно описывает точки каждого кластера их проекциями на несколько внутрикластерных главных компонент. Оптимизация модели ведется циклически: после каждого переразбиения набора примеров на кластеры выполняется расчет главных компонет для каждого кластера, после чего примеры по результатам аппроксимации переразделяются по кластерам, центры которых заново пересчитываются уже для новых разбиений, и т.д.. Подобная же схема построения кусочно-линейного регрессора (с каждым кластером связано свое уравнение регрессии) одновременно была предложена [22].

В [23] дан иной алгоритм идентификации границ и числа областей работы отдельных нейрорегрессоров. А для прогноза скалярного временного ряда предложен способ [24] идентификации скрытой марковской цепи, для каждого из состояний которой строится нейросетевая прогнозная модель.

Очевидно, что обучение набора моделей-экспертов, каждая из которых работает в своей области данных, легко распараллеливается. Но и шаг кластеризации данных, т.е. идентификации этих областей, требует повышения его робастности и возможности выделения кластеров иной формы (путем выбора других гипотез о компактности кластеров по сравнению с гипотезой о смеси многомерных нормальных распределений) – т.е. и здесь можно специально привлекать эффективно распараллеливаемые методы и ставить и решать задачи, слишком трудозатратные при однопроцессорных расчетах.

Отбор информативных независимых переменных

Схема [25] отбора и группировки информативных независимых переменных использует обучение и редукцию структур достаточно большого числа нейросетей.

В [26] предложена схема идентификации информативных признаков и получения координат разделяющих классы поверхностей путем исследования чувствительности значений выходных сигналов нейросети в различных точках пространства независимых переменных – в ходе мелкошагового сканирования области значений в пространстве достаточно большой размерности (все входные сигналы сети, и неинформативные, и информативные, поскольку информативность выявляется как раз по результатам этой процедуры), т.е. является прямой переборной процедурой.

Интересена и парадигма multitask learning [27]: если в предметной области есть несколько задач прогнозирования или классификации, опирающихся на общий набор независимых признаков, то, по сравнению с построением независимого решателя для каждой задачи, одновременное решение всех задач одним решателем (нейросетью с несколькими выходными сигналами) может повысить обобщающие способности, т.к. такая нейросеть строит и использует "промежуточные рассуждения", пригодные для решения всех задач, т.е. потенциально более адекватные причинно-следственным реалиям предметной области и менее подверженные шумам в измерениях значений отдельных независимых признаков. Можно искусственно создавать ситуацию одновременного обучения нескольким задачам путем перевода найденных неинформативных независимых признаков в число зависимых – см. идеи и эксперименты [28]. Отсутствие же в [28] формальной теории и индикаторов требует экспериментов по анализу различных вариантов сочетаний неинформативных независимых переменных, искусственно превращаемых в зависимые.

Заключение

Дан перечень нескольких групп задач, возникающих при нейромоделировании, и методов, внедрение которых в практику повысит эффективность нейромоделирования, но которые при этом требуют значительных вычислительных затрат и поэтому являются кандидатами для параллельных реализаций. Необходимость включения дополнительных сервисных алгоритмов, не обязательно нейросетевых, в состав нейропрограмм и их использования при решении задач обоснована в [29].

Некоторые случаи (например, задачи комбинаторного перебора примеров выборки) наиболее эффективно могут быть решены при запросе программой числа параллельных процессоров и размеров их кэшей и оптимальном, возможно, параллельно-последовательном разделении фрагментов задачи между процессорами с целью поместить все данные, обрабатываемые процессором, в его кэш и получить сверхлинейное ускорение за счет исключения повторных обращений к обычной памяти компьютера. Это еще один способ повышения скорости работы программ, который можно применять при практическом программировании.

Литература

1. *Осовский С.* Нейронные сети для обработки информации. - М.: Финансы и статистика, 2002. – 344с.
2. *Torresen J., Tomita S., Landsverk O.* The relation of weight update frequency to convergence of BP / Proc. World Conf. Neural Networks, Washington, DC, USA. 1995. – pp.679-682.
3. *Wilson D.R., Martinez T.R.* The general inefficiency of batch training for gradient descent learning / Neural Networks. 2003, Vol.16. Issue 10. – pp.1429-1451.
4. *Fukumizu K.* Effect of batch learning in multilayer neural networks / Proc. 5th Intl. Conf. on Neural Information Processing (ICONIP'1998). 1998. – pp.67-70.
5. *Lawrence S., Giles C.L.* Overfitting and neural networks: Conjugate gradient and backpropagation / Proc. IJCNN'2000 Como, Italy. 2000. – pp.114-119.

6. *Wilson D.R., Martinez T.R.* The need for small learning rates on large problems / Proc. IJCNN'2001, Washington, DC, USA. 2001. – pp.115-119.
7. *Riedmiller M.* Advanced supervised learning in multi-layer perceptrons – from backpropagation to adaptive learning algorithms / Computer Standards and Interfaces, 1994. Vol.16. pp.265-278.
8. *Царегородцев В.Г.* Оптимизация предобработки данных: константа Липшица обучающей выборки и свойства обученных нейронных сетей // Нейрокомпьютеры: разработка, применение. 2003, №7. – с.3-8.
9. *Hara K., Nakayama K., Kharaf A.A.M.* A training data selection in online-training for multilayer neural networks / Proc. IJCNN'1998, Anchorage, Alaska, USA. 1998. – pp.2247-2252.
10. *Röbel A.* Dynamic pattern selection for faster learning and controlled generalization of neural networks / Proc. ESANN'1994, Brussels, Belgium. 1994. – pp.187-192.
11. *Щеглов И.Н., Демченко С.А., Богомолов А.В., Подлесских А.А.* Алгоритм формирования репрезентативной обучающей выборки искусственной нейронной сети / Материалы V Всеросс. Конф. "Нейрокомпьютеры и их применение", Москва, 1999. – С.405-407.
12. *Царегородцев В.Г.* Робастная целевая функция с допуском на точность решения для нейросети-предиктора / Нейрокомпьютеры: разработка, применение. 2003, №12.
13. *Hämäläinen J.J., Järvimäki I.* Input projection method for safe use of neural networks based on process data / Proc. IJCNN'1998, Anchorage, Alaska, USA. 1998. – pp.193-198.
14. *Царегородцев В.Г.* Уточнение решения обратной задачи для нейросети-классификатора // Нейрокомпьютеры: разработка, применение. 2003, №12.
15. *Cortes C., Jackel L.D., Solla S.A., Vapnik V., Denker J.S.* Learning curves: Asymptotic values and rate of convergence / Advances in Neural Information Processing Systems 6 (1993). Morgan Kaufmann, 1994. – pp.327-334.
16. *Cortes C., Jackel L.D., Chiang W.-P.* Limits on learning machine accuracy imposed by data quality / Advances in Neural Information Processing Systems 7 (1994). MIT Press, 1995. – pp.239-246.
17. *Gu B., Hu F., Liu H.* Modelling classification performance for large data sets: An empirical study / Lecture Notes in Computer Science, 2001. Vol.2118. – pp.317-328.
18. *Fahlmann S.E., Lebiere C.* The cascade-correlation learning architecture / Advances in Neural Information Processing Systems 2 (1989). Morgan Kaufmann, 1990. – pp.524-532.
19. *Duch W., Jankowski N.* Transfer functions: hidden possibilities for better neural networks / Proc. ESANN'2001, Bruges, Belgium. 2001. – pp.81-94.
20. *Lesoray O., Cardot H.* A neural network architecture for data classification / Int. Journal of Neural Systems, 2001. Vol.11. №.1. – pp.33-42.
21. *Kambhatla N., Leen T.* Fast nonlinear dimension reduction / Advances in Neural Information Processing Systems 6 (1993). Morgan Kaufmann, 1994. – pp.152-157.
22. *Choi I.Y., Kil R.M., Choi C.-H.* Piecewise linear regression networks and its application to time series prediction / Proc. IJCNN'1993, Nagoya, Japan. 1993, Vol.2. – pp.1349-1352.
23. *Lee K., Lee T.* Design of neural networks for multi-value regression / Proc. IJCNN'2001, Washington, DC, USA. 2001. – pp.93-98.
24. *Rynkiewicz J.* Hybrid HMM/MLP models for time series prediction / Proc. ESANN'1999, Bruges, Belgium. 1999. – pp.455-462.
25. *Царегородцев В.Г.* К определению информативности независимых переменных для нейронной сети // Материалы XI Всеросс. семинара "Нейроинформатика и ее приложения", Красноярск, 2003. - 215с. – С.176-177.
26. *Goh T.-H.* Semantic extraction using neural networks modelling and sensitivity analysis / Proc. IJCNN'1993, Nagoya, Japan. 1993, Vol.1. – pp.1031-1034.
27. *Caruana R.* Multitask learning / PhD thesis, School of Comp. Sci., Carnegie Mellon Univ., Pittsburg, PA, USA. 1997.
28. *Caruana R., de Sa V.R.* Benefitting from the variables that variable selection discards / Journal of Machine Learning Research. 2003, Vol.3. – pp.1245-1264.
29. *Царегородцев В.Г.* Взгляд на архитектуру и требования к нейроимитатору для решения современных индустриальных задач // Материалы XI Всеросс. Семинара "Нейроинформатика и ее приложения", Красноярск, 2003. - 215с. – С.171-175.